AMENDMENTS TO THE CLAIMS

1. (Currently amended) A system for sharing secure sockets layer (SSL) sessions across multiple processes comprising:

an application process;

an SSL daemon process;

an SSL wrapper process; and

a plurality of SSL application programming interface (API) calls for communication between the application process and SSL wrapper process, for communication between the SSL wrapper process and the SSL daemon process, and for communication between the SSL daemon process and at least one SSL session.

2. (Currently amended) The system of claim 1 wherein the SSL wrapper process receives ~~a~~ requests for ~~an~~ SSL sessions from an application program, determines a~~whether the~~ request is for a shared ~~or unshared~~ SSL session, passes the requests for the~~a~~ shared SSL session to the SSL daemon process, receives a return code from the SSL daemon process, and passes the return code to the application program.

3. (Original) The system of claim 2 wherein the requests received by the SSL wrapper process include a first input parameter, the first input parameter indicating whether or not a shared SSL session is requested.

4. (Original) The system of claim 2 wherein the SSL wrapper process receives a second input parameter and passes the second input parameter to the SSL daemon process, the second input parameter comprising the data the application process requests secured by an SSL session.

5. (Original) The system of claim 2 wherein the SSL daemon process receives a request for a shared SSL session from the SSL wrapper process, passes requests for a shared SSL session to a shared SSL session, receives a return code from the SSL session, and passes the return code to the SSL wrapper process.

PAGE 3/7 * RCVD AT 12/27/2004 7:33:31 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/3 * DNIS:8729306 * CSID:8602860115 * DURATION (mm-ss):03-48

BEST AVAILABLE COPY

6. (Original) The system of claim 4 wherein the SSL daemon process receives a second input parameter from the application process and passes the second input parameter to the SSL session.

7. (Currently amended) A method for sharing secure sockets layer (SSL) sessions across multiple processes, comprising:

receiving, by at least one SSL wrapper process, ~~receiving~~ a request for a shared SSL session from an application process;

receiving, by an SSL daemon process, ~~receiving~~ at least one request for a shared SSL session from the SSL wrapper process;

calling, by the SSL daemon process, ~~calling~~ at least one SSL session;

receiving, by the SSL daemon process, ~~receiving~~ at least one return code from at least one called SSL session;

receiving, by at least one SSL wrapper process, ~~receiving~~ at least one return code from the SSL daemon process; and

passing by, at least one SSL wrapper process, ~~passing~~ a return code to the application process.

8. (Original) The method in claim 7 wherein a request for an SSL session includes a first input parameter, the first input parameter indicating whether or not a shared SSL session is requested.

9. (Original) The method of claim 7 wherein the SSL wrapper process communicates with the application process using SSL application programming interface (API) calls, the SSL wrapper process communicates with the SSL daemon process using SSL application programming interface (API) calls, and the SSL daemon process communicates with SSL sessions using SSL application programming interface (API) calls.

10. (Original) An article of manufacture comprising:

a computer useable medium having computer readable program code embodied therein for performing a method for sharing secure sockets layer (SSL) sessions across multiple processes, ~~the computer readable program in said article of manufacture~~ the method comprising:

~~computer readable program code for causing a computer to~~ receiving, by an SSL wrapper process, a request for an SSL session from an application process;

~~to~~ determining, by the SSL wrapper process, whether the request is for a shared SSL session or an unshared SSL session;

~~to~~ passing, by the SSL wrapper process, the request ~~for a shared SSL session~~ to an SSL daemon process, when the request is for the shared SSL session;

~~and to~~ receiving, by the SSL wrapper process, a return code from the SSL daemon process, when the request is for the shared SSL session;

computer readable program code ~~for causing a computer to receive at least one~~ request for a shared SSL session

~~to~~ calling, by the SSL wrapper process, an SSL session, when the request is for the unshared SSL session; and

~~to~~ receiving, by the SSL wrapper process, a return code from the SSL session, when the request is for the unshared SSL session.

~~and to pass a return code to an SSL wrapper~~ process.

11. (Currently amended) The article of manufacture of claim 10 ~~further comprising computer readable program code for causing a computer to receive a request for an SSL session~~, wherein the request includes a first input parameter indicating whether the request is for the ~~or~~ not a shared SSL session or the unshared SSL session ~~is requested~~.

12. (Currently amended) The article of manufacture of claim 11 ~~10 further comprising computer readable program code for causing a computer to receive a request for an SSL session~~, wherein the request includes a second input parameter, the second input parameter being the data the an application process requests to be secured by an SSL session.

REMARKS

In response to the Office Action dated September 24, 2004, Applicant respectfully requests reconsideration based on the above claim amendments and the following remarks. Applicant respectfully submits that the claims as presented are in condition for allowance.

Claims 1-12 are pending in this application. Claims 1, 2, 7, and 10-12 are amended. Claims 1, 2, 7, and 10 12 contain no new matter and are supported by the original application, including the drawings and the original claims.

Claims 1-6 and 10-12 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention.

Claims 1, 2, 7, and 10-12 are amended. Claims 1 and 7 are amended to provide antecedent basis for "the SSL daemon process". Claim 7 is amended to correct method claim format so that each element begins with a gerund. Claims 1, 2, 10, 11, and 12 are amended to clarify the subject matter that Applicant regards as the invention.

Claim 1 was rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,657,390 to Elgamal et al. ("Elgamal") in view of an online glossary definition of the term "Stunnel" by Trojnara et al. ("Stunnel").

A *prima facia case* of obviousness under 35 U.S.C. § 103(a) requires that the combination of references teach or suggest all the claim elements. The proposed combination of Elgamal and Stunnel in the Office Action fails to teach or suggest all the claim elements, because it fails to teach or suggest, for example, an application process using SSL API calls to communicate with an SSL wrapper process.

The Office Action states "Elgamal does not teach the SSL wrapper process." (Office Action, page 3, para. 5) Furthermore, Stunnel fails to teach or suggest an application process using SSL API calls to communicate with an SSL wrapper process. For at least these reasons, claim 1 is patentable over the combination of Elgamal and Stunnel, as discussed below.

Claim 1 recites, *inter alia*, "a plurality of SSL application programming interface (API) calls for communication between the application process and SSL wrapper

process". The claimed invention is very different from Stunnel. Stunnel states "The concept is that having non-SSL aware daemons running on your system you can easily setup them to communicate with clients over secure SSL channel." (Stunnel, page 1, description section, second sentence). In other words, Stunnel allows regular server applications, i.e., servers using straight socket APIs rather than SSL APIs, to connect to a local proxy that communicates with remote clients using SSL. As a result, the server application has no idea that SSL is being used. In the claimed invention, the local server applications are SSL aware, because the application process uses SSL API calls to communicate with an SSL wrapper process.

Stunnel uses similar terminology, which is probably a cause of confusion. Stunnel uses the term "SSL encryption wrapper" for something that intercepts socket APIs issued by non-SSL aware applications and then generates SSL APIs that are issued. The so called "SSL encryption wrapper" of Stunnel acts as a proxy that neither the client or server application knows exists. Communication in and out of the server application is standard sockets in Stunnel. Communication between the server node and the remote client is standard SSL in Stunnel. Conceptually, there is a standard socket between the server application and the "SSL encryption wrapper" running in the server and a standard SSL session between the "SSL encryption wrapper" and the remote client. Any data that the "SSL encryption wrapper" receives over the socket is sent over the SSL session and vice-versa. Stunnel is simply one of many ways of implementing an SSL proxy that enables SSL to be used by application that have no knowledge of SSL. In the claimed invention, by contrast, the application process is aware of the SSL session and issues SSL APIs (not socket APIs). Stunnel fails to disclose SSL-aware applications. Therefore, claim 1 is patentable over the combination of Elgamal and Stunnel.

Claims 7, 8, 10-12 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,772,333 B1 to Brendel et al. ("Brendel") in view of Stunnel.

A *prima facie case* of obviousness under 35 U.S.C. § 103(a) requires that the combination of references teach or suggest all the claim elements. The proposed combination of Brendel and Stunnel in the Office Action fails to teach or suggest all the claim elements, because it fails to teach or suggest, for example, a shared SSL session.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: ______________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.